Demonstrating data-driven multivariate regression models for simulated PRACH data using machine learning

Master's Thesis
Emilia Karhumaa
2465290
Unit of Mathematical Sciences
Data Science
University of Oulu
Spring 2020

Abstract

This thesis studies the application of multivariate regression models in simulated PRACH data. The construction of the regression models have been built using the machine learning algorithms Lasso and Gaussian Process Regression with different kernels. I used two different regression algorithms to compare which one fits best. The data contains input and output variables. For each output variable, its own model was constructed using the above-mentioned machine learning algorithms.

The physical random channel (PRACH) handles how the user is connected to the broadband network, more specifically, it requests uplink allocation of the user to the base station. Data has been collected from the 5gNR link-level simulator which have built to simulate radio communication with 5G technology. The simulator compares two different test runs with each other. It is expected that these runs work in exactly the same way. The purpose of the thesis was to investigate the behaviour and divergence of the test runs, and sort important variables in the definition of a random channel.

Keywords: regression modelling, multivariate analysis, machine learning, PRACH, Lasso, Gaussian Process Regression, simulator

Tiivistelmä

Tämä opinnäytetyö tutkii monimuuttuja regressiomallien soveltamista simuloituun PRACH dataan. Regressiomallit on rakennettu käyttäen koneoppimisalgoritmeja Lasso ja Gaussian Prosessi Regressio erilaisilla ytimillä. Käytin kahta erilaista regressioalgoritmia vertaillaksesi mikä sopii parhaiten. Data sisältää selittäviä muuttujia sekä tutkittavia tulosmuuttujia. Jokaiselle tulosmuuttujalle rakennettiin oma malli käyttäen edellä mainittuja koneoppimisalgoritmeja.

Fyysinen satunnaiskanava (PRACH) käsittelee, kuinka käyttäjä saadaan liitettyä laajakaistaverkkoon, tarkemmin sanottuna se pyytää uplink allokointia käyttäjältä tukiasemalle. Data on kerätty 5gNR linkkitason simulaattorista, joka on rakennettu simuloimaan radioviestintää 5G teknologialla. Simulaatori vertailee kahta erilaista testiajoa keskenään. Näiden ajojen odotetaan toimivan täsmälleen samalla tavalla. Tutkielmassa tarkoitus oli tutkia testiajojen käyttäytymistä ja eroavaisuuksia, sekä yrittää nostaa tärkeitä muuttujia esille satunnaiskanavan määrittelystä.

Avainsanat: regressiomallit, monimuuttuja-analyysi, koneoppiminen, PRACH, Lasso, Gaussian Prosessi Regressio, simulaattori

Foreword

I am very grateful that the Nokia Solutions and Networks company gave me a chance to do my thesis work for them. I got to be part of some high-quality radio communication product evolutions. I was part of a system on a chip (SoC) team for most of my time at Nokia and my master's thesis was done for this team.

I would like to thank a lot Pekka Tuuttila and Olga Kayo a lot for supporting me during my master's thesis at Nokia. Tuuttila was my supervisor, and he gave me good answers to my questions and guidance about radio communications systems. With Tuuttila we created structure for my thesis. Kayo provided me with support for technical systems in Nokia and instructions for my thesis with important opinions. Many thanks to Mark Griffiths for helping me write a more understandable English text.

From the University of Oulu, I would like to thank my supervisor Erkki Laitinen a lot. With him I found my way to the Nokia company, and I am very grateful for that. He was also always ready to provide guidance during my thesis work. Laitinen gave me a lot of good material and advice about mathematical part of this thesis work. With this guidance I was able to proceed with writing the thesis at a good pace. Also, I want to thank Leena Ruha and Esa Läärä as being the other support persons from the university.

This master thesis work has received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 737494. This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation program and Sweden, France, Spain, Italy, Finland and the Czech Republic. The purpose of this project is to create a framework incorporating methods for continuous validation and devaluation in model-based methods. These methods can provide benefits for improved quality and productivity of large and complex systems. [20]

Abbreviations and Symbols

Abbreviations

DL	Downlink
flp	floating-point
fxp	fixed point
gNB	NR NodeB
GP	Gaussian Process
GPR	Gaussian Process Regression
Lasso	Least absolute shrinkage and selection operator
LL	Link-Level
LLE	Local Linear Embedding
LSR	Least Squares Regression
MSE	Mean Squared Error
MVA	Multivariate analysis
MVRA	Multivariate regression analysis
NG	Next Generation
NR	New Radio
PCA	Principal Component Analysis
PRACH	Physical Random Access Channel
RAN	Radio Access Network
RMSE	Root Mean Squared Error
RRC	Radio Resource Control
Rx	Receiver
SGD	Stochastic Gradient Descent
SoC	System on a Chip
SVC	Support Vector Classifier
Tx	Transmitter
UE	User Equipment (mobile phone)
UL	Uplink
VBGMM	Variational Bayesian Gaussian Mixture Model
5G	5th Generation of Cellular Wireless Standards

Symbols

β	Regression coefficient matrix
$C(Y\|X)$	The property of conditional distribution of Y for given X
δ	Delta
$E[Y\|X]$	The conditional expected value of Y for given X
$g(X)$	A distribution of given X
H	A vector of explicit basis functions $h(x_i)^T$
$K(x, x')$	A kernel function
L_1	Constraint of L_1-norm regularization
M	Number of input variables
N	Number of samples
$P(Y\|X)$	The conditional probability of Y for given X
R	Number of output variables
$r^{(j)}$	Element j of vector r
σ^2	A noise variance
θ	A hyperparameter
X	Input matrix
x_{ij}	Element of input matrix
Y	Output matrix
y_i	Element of output matrix
$\|\cdot\|$	A absolute value
$(\cdot)_p$	A predicted value
$\|\cdot\|_1$	1-norm
$\|\cdot\|_2^2$	Euclidean norm
$\langle\cdot,\cdot\rangle$	Scalar product
$\{(x_i, y_i)\}_{i=1}^N$	A set of variable pairs
$x_i = (x_{i1}, \ldots, x_{ip})$	A vector of features with p-dimensional
$\{X_t\}_{t\in T}$	A family of random variables

Contents

1 Introduction

1.1 Background

This thesis deals with the quality development of a part of 5G base station as a product. Figure 1 shows product development chain by very simplified steps.

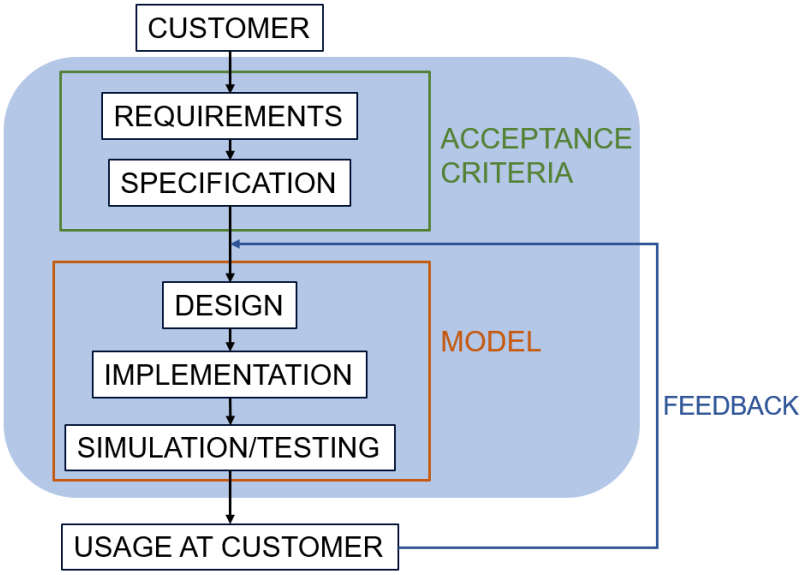


Figure 1: Product development chain

Product development is started from customer. Customer needs to defines his needs for product. Product vendor defines requirements and specification for product. These two definitions construct acceptance criteria for the products. To my thesis Nokia is the customer and many acceptance criteria have been defined.

When requirements and specification steps have been defined, we are able to start to make a model to the product because we know what customer wants. Model will be simulator of product. This contains three steps, which are design of the simulator, the implementation of this design and simulation of the model. When simulation is done, then the model needs to be verified by number of test cases that demonstrates how simulation of product works.

When we have version of product as model which can handle all acceptance criteria, we are ready for the last step. It is product usage by customer. In this step it is very important that customer gives feedback of usage that is used to improve product more.

My role in this chain was to make a simple uplink functionality model in the simulator that it is easy to compare its behaviour against acceptance criteria. In figure 1 my role is shown as a blue area in product development chain.

More specific, PRACH data was generated by the simulator. The simulator is a model of a product. We have three research questions regarding on that PRACH data. First, we are interested in that how input variables influence to output variables. Second, we wanted to know more precisely that which input have strongest effect to each output. Lastly, we are interested in outputs which get the worst results. If we can find out answers for those questions, we can concentrate on those sorted variables and then try optimizing them in simulator.

This multidimensional problem was proposed by customer to be handled by machine learning algorithm. The influence between inputs and outputs could be solved by regression model between them. It also allows to study and compare variables in data. It could be possible to do that with two different kind of regression algorithm to compare which one is better and then used that algorithm for observing influences. Those methods are Lasso regression and Gaussian Process Regression (GPR). Lasso is a linear regression method and GPR is one of the stochastic methods in regression modelling. Difference between these methods is that Lasso builds a linear combination of inputs and GPR does that of outputs in data. [6]

1.2 Simulator

The simulator is 5gNR link-level (LL) simulator which have built to simulate radio communication with 5G technology. *"LL simulation is basically a software implementation of one or multiple links between the Tx and the*

Rx, with a generated channel model to reflect the actual transmission of the waveforms from Tx to Rx. −−− In the LL simulator, we get the mapping curves of BLER and Signal-to-Noise-Ratio (SNR), which are used as a baseline for measuring the performance of the LL simulation. [14]" The signal propagation go through various physical channel models. The simulator performs both downlink (DL) and uplink (UL) transmission. [25] Purpose of the simulator is to study and develop algorithms to be used in 5G base station. These algorithms in simulator is built with Matlab.

Simulator is built so that at first, we define some input variable values. With these defined inputs, simulator communicate as User Equipment (UE) and base station over broadband radio channel. Simulator does two different test runs in every test. The first run is floating-point (flp) test and the second is fixed point (fxp) test. The difference between the two tests is the decimal place in the numerical representation. The flp test is optimal and it uses exact numerical values. The fxp has a limited number of bits available to represent numerical values and it specifies the decimal precision. Input variables are same for both flp and fxp test runs. We get results of this communication. These measured result values show how successful this communication is. Both runs give their own results.

We can test simulator's behaviour by changing those input variables. We collect data from each test and the purpose is to do it with most interesting combinations of input variables. In addition, we can perform this test as many times as we like and thus increase the result data volume.

The simulator is a very complex system and part of it is the case study of the thesis. This case explores the Physical Random Access channel (PRACH) in simulator with some constant input variables.

1.3 PRACH

Physical Random Access Channel (PRACH) sets the connection between a UE and gNB, it is used in the first step of PRACH procedure [26]. More specific, there is a couple of main purposes to PRACH. It must achieve syn-

chronization between UE and gNB. gNB is logical NG-RAN node which providing NR user plane and control plane protocol about terminations towards the UE. In this case it will be handled by UL transmission. The synchronization process happens only when needed and should be dedicated only for a one specific UE. PRACH handles connection request from Radio Resource Control (RRC). In beamforming case, PRACH indicates to gNB that it finds out the initial best beam. [18]

By simplified words, PRACH handles how UE can be connected to the broadband network. Figure 2 illustrates the allocation between base station and UE. The uplink allocation means that UE send first message to the base station.

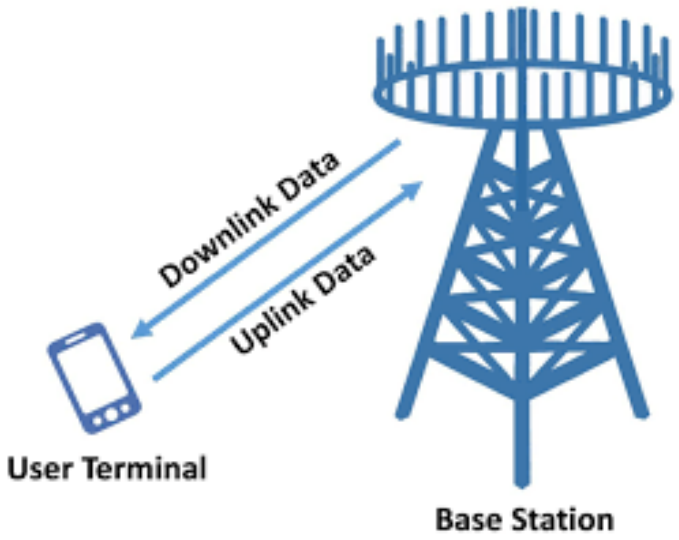


Figure 2: Uplink and downlink communication between base station and UE. [24]

1.4 Data

Data is simulated using the simulator described in subsection 1.2. This data is multidimensional and there are many input and output variables. Those inputs are variables which create network scenario to base station and re-

ceiver. Outputs are results of this communication.

At first data is parsed from diary. This diary is console output of test which have been executed in simulator. Data has been parsed from general Matlab file format to matrix format. In data matrix columns are variables and rows are samples. These samples contain different cases about the input variables so that each row has outcome from simulator. Cases could be similar with each other too, but results still can be different.

I have chosen some of the input variables to be constant for PRACH in this thesis. Precisely in data there are $M = 9$ input variables, the values of which are changed between the test rounds. Seven of these nine variable values have exact value range. At the beginning of test, simulator does random choice from this range and set picked value to be variable value in this test round. The first of other two variables depends on other input's value and then gets only 0 or 9 to be possible value. The second is alternately 0 or 1. Value choosing has been done to all nine input variables in every test round in data.

I have chosen $R = 14$ output variables. As described in subsection 1.2 Simulator, flp and fxp tests give their results of each test round. For twelve outputs, there are two different values, flp and fxp. Two other output values are only available for flp test. For those outputs which have flp and fxp values, the main purpose is that these values should be exactly same. I have modified data so that it has delta δ value for those outputs. Delta (Definition 1.1) is absolute value about difference between flp value and fxp value.

Definition 1.1. Delta for $y_{ij,flp} \in Y_{flp}$ and $y_{ij,fxp} \in Y_{ij,PRED}$

$$\delta = |y_{ij,flp} - y_{ij,fxp}|$$

where $j = 1, \ldots, R$ and $i = 1, \ldots, N$.

1.4.1 Data format

Let the number of input variables to be M and the number of samples to be N. Then the dimension of the input matrix X is N x M. Input matrix form

is

$$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1M} \\ x_{21} & x_{22} & \dots & x_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{NM} \end{bmatrix}.$$

The output result matrix Y consists of R columns containing N samples.. Each column is one of the outputs. Let the number of output results to be R and the number of samples to be N. Then the dimension of output matrix Y is N x R. The column Y_i in the matrix Y has the form

$$\begin{bmatrix} y_{i1} \\ y_{i2} \\ \vdots \\ y_{iN} \end{bmatrix}$$

where $i = 1, \dots, R$.

Our goal is predicting regression model between those two matrices. Let f_i be mark for the model of output Y_i and the form is

$$f_i(X) = Y_i, \quad i = 1, \dots, R.$$

We built own regression model for each output Y_i.

Data format is same for the training and the testing data. Only number of samples N is different for those data. The training data is about 70% of whole data and rest of it belongs to the testing data.

1.5 Scope of thesis

In Chapter 2, the theory of multivariate analysis and regression models are presented. Chapter 3 presents machine learning and its lalgorithms. Chapter 4 describe the general view and theory of regression method Lasso. In Chapter 5, the theory and mathematical notations of Gaussian Process Regression are presented. Chapter 6 demonstrate the method validation for data. This

chapter also presents the results of the validation and discusses the success of the demonstration. Chapter 7 summarize the content of thesis and the results of the analysis against the research questions are compared. Chapter 8 consider possibilities of new methods for simulated PRACH data in future.

2 Multivariate regression

Multivariate aspects mean that there are multiple variable for all features in data to be analyzed. Number of target values can also be more than one. When we have that kind of situation in our data, it brings difficulties to get right information from data and more variation for analysis. This situation needs to be handled by multivariate analysis (MVA) so that we get enough information from data. One of the most important things in MVA is that it takes account of correlation between variables.

2.1 Multivariate analysis

Multivariate analysis (MVA) means that data which need to be analyzed have multiple parameters or measurements. MVA analyses those variables together. In each MVA method, we can use for example correlations of variables to study their relations between each other. The techniques in MVA are more valuable with correlated variables. Others common key statistics are covariance and variance. [5, 15]

Multivariate analysis would need preprocessing of data before it can be analyzed. There can be a lot of different types and value ranges between variables. So, all variables need to be set to the same position to ensure that the results are on the same level. Preprocessing can be standardization or normalization. By those methods the values of columns can be changed to a common scale. Other preprocessing thing are cleaning, editing and wrangling data.

The purpose of MVA is to get information extraction of multidimensional data. Information could be for example classes, correlations or variance of variables. Other thing is data reduction which could also be the main purpose. If data have multiple dimension, and we want informative visualization about it, then we need to use MVA techniques for data reduction. Data reduction could be used also for the labelling data.

Methods of MVA could be for example Principal Component Analysis

(PCA), Local Linear Embedding and Least (LLE) absolute shrinkage and selection operator (Lasso) regression. These multivariate analysis methods are very commonly used in machine learning. We also can have one output or multiple outputs which affect to method selection. When choosing the methods, you need to know your data and the goal of your analysis.

2.2 Regression models

In regression models we have input X and response value Y [21]. A regression model is a connection between inputs and output. The connection is handled using a vector of coefficients. For a basic linear regression model the vector of coefficients is $\beta = [\beta_0, \beta_1]$. There β_0 is the parameter, which is not containing any X_i. The β_1 is weight to handling input X in regression model. The model basic form is

$$f(X) = Y.$$

As open formula the form of the basic linear regression model is

$$Y = \beta_0 + \beta_1 X.$$

But almost always the situation is more complicated. Regression model form depends on the property of the distribution of response Y for given X. Let that be $C(Y|X)$. This can be for example the conditional expected value $E(Y|X)$ or the conditional probability $P(Y|X)$. Now the general form for regression model is

$$C(Y|X) = g(X).$$

In linear regression model we can assume that $g(X) = X\beta$. Then the linear regression model is

$$C(Y|X) = X\beta.$$

Non-linear regression uses equation which is non-linear. For given set X the model form of response values Y is

$$Y = f(X, \beta) + \epsilon$$

where ϵ is an error term, β is a vector of coefficients and f is a known regression function. We can say that if the form of model is not in linear format then it is non-linear model. [17]

2.2.1 Multivariate regression models

Multivariate regression analysis (MVRA) estimates regression model for data which have more than one input variables and possibly also more than one output variables. In this method we are interested about how to predict Y by input values of X. The other purpose is to find out variable's correlation between each other and how they are related as together to the output Y. Aspect of interest is also to choose some input variables to predict this model, meaning that we take for analysis only those input variables which really are significant. Lasso regression is one of the applications in MVRA. [8, 15]

In multivariate regression models our input is M-dimensional vector $X = [x_1, x_2, \ldots, x_M]$ and target value is Y or vector $Y = [y_1, y_2, \ldots, y_R]$ where R is number of the target variables, outputs. Let $\beta = [\beta_0, \beta_1, \ldots, \beta_M]$ be the vector of coefficients. There β_0 is the intercept value and other coefficients β_i are weight to handling each input x_i in regression model. Now for example, the formula of linear multivariate regression model $f(X)$ is

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_M x_M$$

where $i = 1, \ldots, R$.

2.3 Optimization

In general optimization means that we want to improve our results. We are interested on which variables have strongest effect to output values and what are the best values to these variables to get as good result as possible. In other words, we try to optimize coefficients for each variable in model.

Some of multivariate analysis methods optimize automatically by reducing number of input dimensions. The purpose is to modify only those variables which explain most of the variances to model.

In multivariate analysis we optimize multivariate function. The multivariate function has several input variables and a scalar output. There is two kind of multivariate optimization methods. First group includes methods which emphasize function evaluations and second represents methods which use information of function's derivative. [13]

The idea of the optimization is usually to find minimum or maximum values of functions. The second goal could be to find the vector of coefficients. With those values we can for example minimize the average absolute error. [16]

Two important and general procedures are to choose a starting point and direction of search. Then move in that direction until we find the optimal point. A successful solution requires good convergence of algorithm. [13, 16]

3 Machine Learning

Machine learning is part of the data science. There is a lot of ways to do machine learning and algorithms are very different between each other. It depends on a lot of data and the purpose of analysis that which learning algorithms should use. [1]

Machine learning algorithms need to learn the data. Learning can be for example model between inputs and outputs, data structure, the most explanatory variables or classification algorithm. Algorithm needs to learn something useful from data that can be used for analysis or prediction. [2]

The machine learning task can be described that how the algorithm of machine learning system should process the data. The data is a collection of samples or features. These features have been collected or measured from specific object of the system under analysis. [2]

In regression task, the machine learning system needs to learn the function between inputs and outputs. Now inputs are features and outputs are numerical values. The purpose is to find the effect of the input variables on the outputs. The algorithm is asked to predict these outputs by using given inputs.

Data in machine learning is a collection of samples. In some methods this data is separated to training and testing data. The purpose of training data is to learn wanted task from data. Testing data investigate how well the task have been learned. [2]

3.1 Data preprosessing

Before using some machine learning algorithm, there is need to prepare the data for it. At first, you must look at your data and try to understand it. For example, you need to find out what kind of variables you have there, what is their type and possible values. Also need to know number of samples and if data is labelled or not. [1]

So, you need to know your data first. Meaning such of things like average,

medians and correlations. You can also make a few simple visualisations of variables and results. These can be box plots, density plots or scatter plots.

When you know your data enough good, then you need to clean it. Cleaning means that need to handle missing values and outliers in data. Also, you must find out that does the data needs to be aggregated.

After cleaning, the next step is to augment the data. Meaning that need to prepare data for modelling. This is method to rescale variables and reduce data dimensionality or even capture more complex situations.

3.2 How to choose machine learning algorithm

When choosing the method, you need to categorize the problem of inputs and outputs. Learning methods can be unsupervised or supervised learning. Unsupervised learning is machine learning methods which try to find out patterns and relationships from data without using any additional information but data itself. These methods can be used for problems which have no knowledge about the effect of variables or data structures. Supervised machine learning methods try to build predictive models about data by using labels provided along with the data. These models will be used for making predictions of the new data sets. In that method there should be labelled data. [19]

There are four big groups of algorithms; classification, clustering, regression and dimensionality reduction. Usually classification and regression methods are supervised learning algorithms. Then dimensionality reduction and clustering are categorized to unsupervised machine learning methods. [1]

When we do not want to predict a category but to quantity them, then we use some regression methods like Stochastic Gradient Descent (SGD) or Lasso regression. To choose methods of regression, we need to know number of samples and if there is target to that some features are more important than other. By regression method we try get function between features and outputs. Regression methods use a training data to learn the model and to

evaluate data to test how good the model fits to data. The training data and testing data are a part of whole data to be analyzed.

After the selection of method, it is needed also consider other methods. It is recommended to use several methods and test their suitability. You can measure the goodness of the algorithm by value of accuracy, explainability, amount of needed preprocessing and how scalable the model is. You need to model so that it answers the questions that you wanted to find out at the beginning, what was the task that the machine learning algorithm was purpose to learn.

3.3 Quality

Quality of machine learning models can be illustrated with some general notions.

Overfitting means that model learn the training data too well. More specific, the model learns so specific informations from training data and then cannot predict the testing or validation data well.

In **Underfitting** situation, the model does not learn the training set well enough. In that case, it does not usually fit well for testing data either.

Capacity of model is the ability of how it fits to data. If the machine learning systems is learning function, a wide variety of that function need to fit. High capacity means that model fits well for both training set and testing data set. Low quality struggle with that. Good capacity leads to the model which is not overfitting or underfitting.

Estimation of each parameter in some parametric model leads to the solution to fit better. Parameter estimation is usually used in regression method. Estimation means that each parameter is optimized for its best value so that then the model fits better.

Bias of an estimator is measured as difference between expected estimation value over the data and the true value of estimator defined as the data-generic distribution. For good model the bias is zero.

Variance of an estimator and the standard error is a measure of how expected estimate from data varying to the true value. Low variance leads to good model. The standard error is the square root of variance.

Error of models can be the training error or testing error. Those errors are usually measured by the mean squared error (MSE) of the estimates. The MSE measures the squared expected deviation between the estimator and the true value. Model has better quality if the MSE value is small. In case of overfitting, the training error is small, but the testing error is big. When model underfitting, the training error is big. Optimal capacity is the point where both bias and variance have together small error value.

Consistency means that when number of features in data increasing then expected value of estimator converging to true value of estimator. If that is true even almost then the bias of estimator will decrease when features of data increase.

4 Lasso Regression

The Least absolute shrinkage and selection operator (Lasso) regression is one of the linear regression methods and it is used to shrinkage regression coefficients. Means that data values shrunk towards a point of central, mean for example. Lasso method performs variable selection and regularization to enhance prediction accuracy and interchangeability of the statistical model. [12]

Lasso regression is related to the Least Squares Regression (LSR) but there has added penalty for the formula. This penalty is added also for Ridge regression, but the constraint is different from those used in Lasso method. With penalty, it is possible to reduce variance of estimate which can be large for the least-squares estimate. This can improve prediction accuracy of estimates, even the bias would increase. [22]

Lasso combines the least-squares loss with an L_1-constraint. In other words, Lasso performs regularization of L_1. This L_1-constraint affect to the coefficients and the settings of the non-informative features to zero. By that Lasso method is automatically featured selection model. This model can be used for large problem and even convex optimization problems. [12, 22]

The Lasso problem is quadratic convex problem and can be handled with Lagrangian form. One of the benefits of Lasso is that it reduces input variables dramatically without sacrificing accuracy. Lasso regression can be used also for classification in machine learning. [22, 23]

The following sections is based most to the reference [22].

4.1 Lasso regression as machine learning algorithm

Lasso regression belongs to supervised regression methods in machine learning algorithms. Regression method is used when the task for the machine learning system is to learn a model, in that case a function, between input variables and target values. From data is not wanted to predict a category but a quantity.

When using Lasso as the learning algorithm, it is needed to split data to training and testing part. This splitting is helpful for analyzing how good is the model's quality and capacity.

Lasso regression in machine learning is very useful for multidimensional data where dimensions can be very big. Lasso is useful also for analyzing input variables, how important they are for explaining the target values. Lasso is doing automatically dimensional reduction with its loss penalty. By this penalty the model selects only those input variables to model which are the most significant to explain target value. The penalty set variables coefficient to zero if it is not useful to predict target value. If the coefficient of the variable is zero, the variable does not affect the target value.

When Lasso regression has learned the model from the training data, then the learned function between inputs and outputs can be used for predicting other data. This other data can be the testing data which can be used for analyzing how good the function fits to that data. The other possible data for prediction can be new dataset. New dataset needed to have those input variable values that the model is selected to function to measure target value for new dataset.

4.2 Mathematical form for Lasso problem

Let N to be set of variable pairs $\{(x_i, y_i)\}_{i=1}^{N}$. Here each $x_i = (x_{i1}, \ldots, x_{iM})$ is vector of features with M-dimensions and y_i is the response value from model. Now the model for the expected value of Y is

$$f(x_i) = \beta_0 + \sum_{j=1}^{M} x_{ij}\beta_j \tag{4.1}$$

where β_0 is an intercept term and $j = 1, ..., M$ are coefficients for each x_i. Model try predicting the response value using linear combination of features.

In Lasso regression the purpose is to solve the following optimization problem

$$\min_{\beta_0,\beta_1} \left\{ \frac{1}{2N} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{M} x_{ij}\beta_j)^2 \right\} \tag{4.2}$$

$$\text{subject to} \quad \|\beta\|_1 = \sum_{j=1}^{M} |\beta_j| \leqslant t$$

where t is user-specified parameter and $\|\beta\|_1$ is L_1-norm of β. User-specified parameter t is a budget on the total L_1-norm. Lasso is finding the best fit and estimate for the pairs (β_0, β) within this budget. [16]

Now we can define X as a matrix with dimensions N x M and Y is vector $(y_1, \ldots, y_N)$ which length is N. Then the problem (4.2) can be written as

$$\min_{\beta_0,\beta_1} \left\{ \frac{1}{2N} \|y - \beta_0 1 - X\beta\|_2^2 \right\} \tag{4.3}$$

$$\text{subject to} \quad \|\beta\|_1 \leqslant t$$

where is $\|\cdot\|_2$ Euclidean norm and 1 is vector of N ones. Notice that matrix X should be normalized to center with 0 and variance 1. The purpose of normalization is that the solution is not dependent on the units of features. This means that

$$\frac{1}{N}\sum_{i=1}^{N} x_{ij} = 0 \quad \text{and} \quad \frac{1}{N}\sum_{i=1}^{N} x_{ij}^2 = 1. \tag{4.4}$$

Mean of the response values y_i are also expected to be centered to zero.

Lasso regression optimization problem can be written as Lagrangian form. The optimization problem (4.3) can be re-expressed as

$$\min_{\beta \in R} \left\{ \frac{1}{2N} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\} \tag{4.5}$$

where $\lambda \geqslant 0$. Now each t where $\|\beta\|_1 \leqslant t$ is true, the corresponding value of λ yields the same solution form as the Lagrangian form. The factor $\frac{1}{2N}$ in formula is not so absolute even can be also 1 or $\frac{1}{2}$.

4.3 Cross-Validation

The user-specified parameter t correspond to the results. Big t allows model to overfitting with training data. Small t fit less to the training model and leading to sparser. So, we need to find optimal t between these two cases

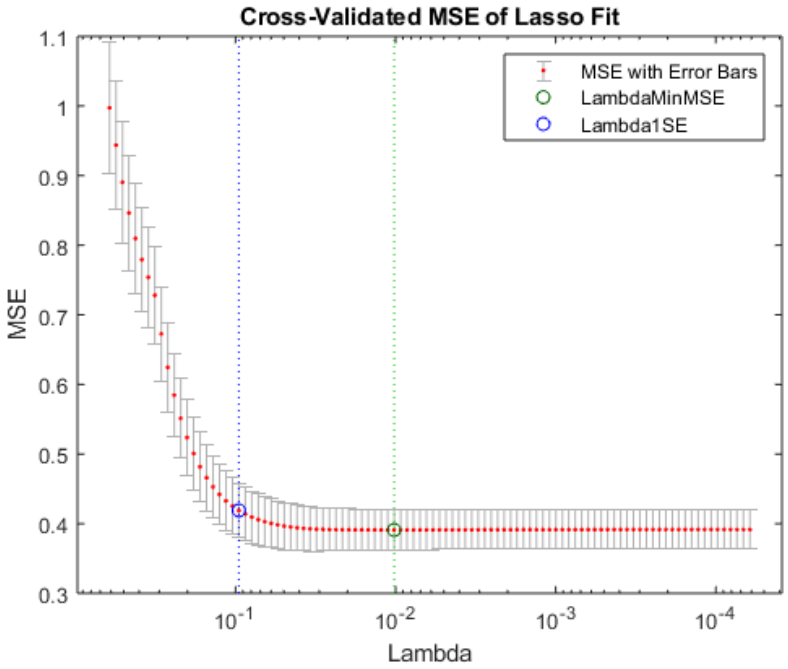


Figure 3: Example of Cross-Validation for Output 1 with K=10.

that we get the best possible Lasso regression results from training data. Otherwise, the prediction error increase to high.

We can find the optimal t by cross-validation which splits given dataset randomly to K multiple training set and by estimating the performance for each set. There need to be more than one training set, i.e. $K \geq 1$ an also $K \leq N$. Usually the chosen value of K is between 5 and 10. At first, we apply Lasso to the training set with different value of t and then each fitted model get own value of prediction error. This is done to every training set. At the end the prediction error is average of each t, called as a $cross - validation\ error\ curve$.

We can plot cross-validation curve. Figure 3 shows example of that. In plot, we have done cross-validation for Output 1 in data with $K = 10$. In addition, we can see point as a green circle when we have minimum cross-validation error for the lambda and alternatively point as a blue circle when there is minimum cross-validation error plus one standard deviation for

lambda. Usually we are selecting K which have minimum cross-validation error.

4.4 Alternative penalties

Lasso penalty is constraint which have added to the Least Squares Regression formula. This penalty is added to reduce variance of estimates. Also, this L_1-constraint affects to the coefficients and setting non-informative features to zero. Formula of constraint is

$$\|\beta\|_1 = \sum_{j=1}^{M} |\beta_j|. \tag{4.6}$$

This constraint is different for Ridge regression which is using L_2-norm as penalty. This Ridge penalty can be written as

$$\|\beta\|_2 = \sum_{j=1}^{M} \beta_j^2. \tag{4.7}$$

There is a couple of possible penalty formula that can be used. The most commonly used are Elastic Net, Group Lasso and Overlap Group. For example, the Elastic Net is using a compromise between Lasso and Ridge penalties. The Elastic Net takes better account of highly correlated variables. In formula, there is parameter α which can be varied between range $[0, 1]$. Now penalty is

$$\frac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j|. \tag{4.8}$$

We can see from the formula that if $\alpha = 1$, the penalty is then Lasso penalty and if $\alpha = 0$, the penalty is Ridge penalty. When using Elastic Net penalty (4.8), the formula of convex optimization problem (4.5) is

$$\min_{(\beta_0,\beta_1)\in\mathbb{R}\mathrm{x}\mathbb{R}^M} \left\{ \frac{1}{2}\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{M} x_{ij}\beta_j)^2 + \lambda \left[\frac{1}{2}(a-\alpha)\|\beta\|_2^2 + \alpha\|\beta\|_1 \right] \right\}. \tag{4.9}$$

Figure 4 shows the constraint regions for these three explained penalties.

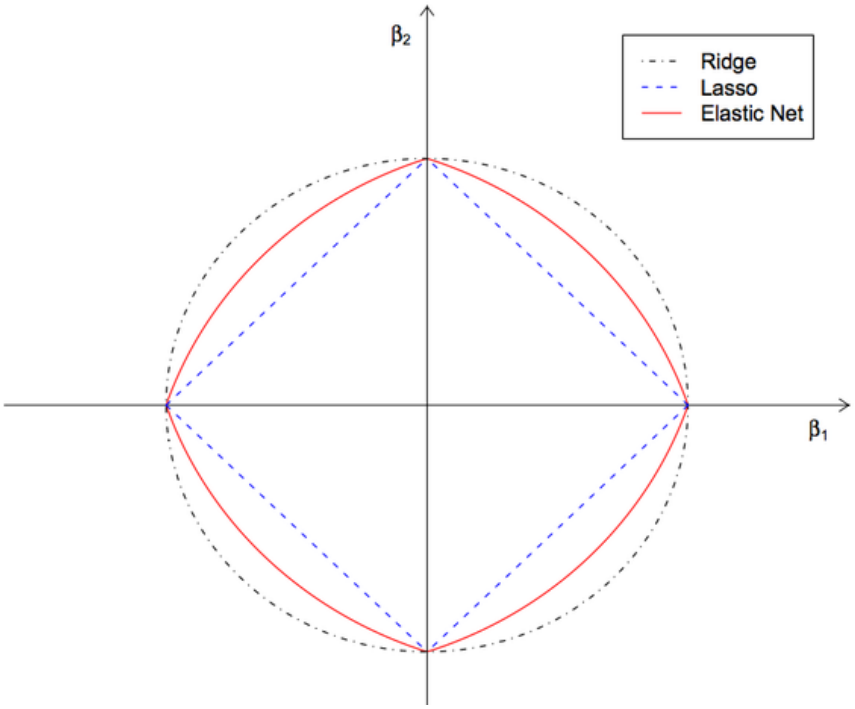


Figure 4: The constraint regions for different penalties. [3]

4.5 The computation of Lasso problem

When solving the optimization problem (4.5), the theory of convex analysis leads to a solution of the form

$$-\frac{1}{N}\langle x_j, y - X\beta\rangle + \lambda s_j = 0 \tag{4.10}$$

where $j = 1, \ldots, M$ and s_j is a subgradient for the absolute value function. If $\beta = 0$ then $s \in [-1, 1]$, otherwise $s = \text{sign}(\beta)$. Then equation (4.10) can be re-expressed as

$$-\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{M} x_{ij}\beta_j\right)x_{ij} + \lambda s_j = 0. \tag{4.11}$$

The intercept β_0 is not penalized and can be omitted when calculating other coefficients β_j. The covariate matrix X and the response y_i are centered to their means. Let the partial residual be as $r_i^{(j)} = y_i - \sum_{k\neq j} x_{ik}\hat{\beta}_k$. This

partial residual leaves only the outcome from j^{th} predictor and removes other. When solving equation (4.11) using this residual, the solution of β_j is

$$\hat{\beta}_j = \frac{S_\lambda\left(\frac{1}{N}\sum_{i=1}^{N} r_i^{(j)} x_{ij}\right)}{\frac{1}{N}\sum_{i=1}^{N} x_{ij}^2} \tag{4.12}$$

where S_λ is the soft-thresholding operator. The operator is presented in section 4.5.1 Soft Thresholding.

If variance of variables is set to have unit value when standardizing those variables, then we can use $\tilde{\beta}$ as the partial residual on variable j in formula. Then solution is

$$\hat{\beta}_j = S_\lambda(\tilde{\beta}). \tag{4.13}$$

When other coefficients are optimized, then we can calculate optimal value for intercept β_0. This value is calculating with following formula

$$\hat{\beta}_0 = \overline{y} - \sum_{j=i}^{M} \overline{x}_j \hat{\beta}_j. \tag{4.14}$$

4.5.1 Soft Thresholding

Assume that we have samples $\{(z_i, y_i)\}_{i=1}^{N}$ where z_i is renamed x_{ij}. Then formula of problem (4.5) is

$$\min_{\beta}\left\{\frac{1}{2N}\sum_{i=1}^{N}(y_i - z_i\beta)^2 + \lambda|\beta|\right\}. \tag{4.15}$$

If case would be univariate optimization, then this minimization would be done by taking the gradient subject to β and setting that to the zero. But now the situation is more complicated and there is multivariate problem. The second complication is that the absolute value of β have not situation that derivative of it would be zero. Now we can find estimation of β as function

$$\widehat{\beta} = \begin{cases} \frac{1}{N}\langle z, y\rangle - \lambda, & if \quad \frac{1}{N}\langle z, y\rangle > \lambda, \\ 0, & if \quad \frac{1}{N}|\langle z, y\rangle| \leq \lambda, \\ \frac{1}{N}\langle z, y\rangle + \lambda, & if \quad \frac{1}{N}\langle z, y\rangle < -\lambda, \end{cases} \tag{4.16}$$

for univariate case.

Now we can define $S_\lambda(x)$ be a soft-tresholding operator

$$S_\lambda(x) = \text{sign}(x)(|x| - \lambda)_+ \tag{4.17}$$

and add it to the formula (4.16) like

$$\widehat{\beta} = S_\lambda \left(\frac{1}{N} \langle z, y \rangle \right). \tag{4.18}$$

Soft-thresholding operator (4.17) translates its input to zero if $|x| \leq \lambda^3$. Otherwise it towards to zero by the amount of λ.

When we are solving the multivariate problem (4.15), we need to update all coefficients β_j as a step. When minimizing the one coefficient then others will stay at their current values. Now each coefficient is updated by formula

$$\widehat{\beta}_j = S_\lambda \left(\frac{1}{N} \langle x_j, r^{(j)} \rangle \right) \tag{4.19}$$

$$\widehat{\beta}_j \leftarrow S_\lambda \left(\hat{\beta}_j + \frac{1}{N} \langle x_j, r \rangle \right) \tag{4.20}$$

where $r_i = y_i - \sum_{j=1}^{M} x_{ij}\beta_j$ are the residuals. When λ is set to zero then problem (4.15) is handled as least-squares problem. By cross-validation method (described in subsection 4.3 Cross-Validation) we are able to find optimal value for λ which is based on data. Anyway, we can set max value for λ and it is $\lambda_{max} = \max_j |\frac{1}{N} \langle x_j, y \rangle|$.

5 Gaussian Process Regression

Gaussian process regression (GPR) using the Gaussian probability distribution which describes random variables. These random variables can be vectors or scalar. The process in GPR is a stochastic process (Definition 5.1). The main thinking behind the GPR is that a function is a very long vector, approximately finite. Then for the function value $f(x)$, where x is the input variable is specified by each entry in that vector. GPR is based on the kernel probabilistic and GPR models are non-parametric. [6]

Definition 5.1. A stochastic process is a family of random variables $\{X_t\}_{t\in T}$, where T is a subset of $[0, \infty)$. A stochastic process can be also named as a random process or a continuous-time process. [9]

The following sections is based most to the references [4], [6] and [7].

5.1 GPR as machine learning algorithm

GPR is supervised learning algorithm to machine learning. GPR belongs to regression methods and target of the method is finding out the predictions of quantities. The purpose is to build a regression model between inputs and outputs values.

One of the benefits of GPR in machine learning is that it could use different kernels to learn the regression function. The chosen kernel need to base on the data distribution. Because the GPR use the distribution of data to build the model, it is needed to be careful that the model do not overfit with the training data. That can be handled with parameter estimation of model and kernel specification.

If dimension of input variables is very large, there is needed to preprocess the data and use for example PCA to reduce dimensionality. When doing that it decreases the possibility of overfitting and time of building model.

Like any other regression models, the GPR method need also data splitting and analysis of data quality and capacity. If it is possible to build model which fits well to data, it can be useful when predicting target values for new data set.

5.2 Mathematical form of GPR

Let $X = \{x_1, x_2, \ldots, x_N\}$ be the set of input variables and let $f(x_i)$, where $i = 1, \ldots, N$, be the response variables. Form of linear regression model is

$$f(x_i) = x^T\beta + \epsilon \tag{5.1}$$

where $\epsilon \backsim N(0, \sigma^2)$. The coefficients β and the error variance σ^2 are estimated from the data. Now response variables are from the Gaussian process (GP). GP is a set of random variables and it leads to that response values are also random variables. Meaning that any number of variables which are finite have a joint Gaussian distribution. Now we can say that for given set of variables X the joint distribution is Gaussian for the response variables $f(x_i)$.

Let a mean function of GP be $m(x) = E[f(x)]$ and a covariance function of GP be $k(x, x')$. Now $k(x, x')$ is a set of kernel parameters. Because we discuss now about the GP, we can say that

$$\text{Cov}[f(x), f(x')] = E[\{f(x) - m(x)\}\{f(x') - m(x')\}] = k(x, x'). \tag{5.2}$$

Let $h(x)$ be feature transform function. This function transforms vector x from original space $\mathbb{R}^M$ to new feature vector space $\mathbb{R}^p$. Function $h(x)$ can be defined as a set of basis functions. Meaning that β is coefficients vector of basic functions with dimensions p-by-1. Let $f(x) \backsim GP(0, k(x, x'))$. Then we can consider model as

$$Y = h(x)^T\beta + f(x). \tag{5.3}$$

Now we can model response's Y instance as

$$P(y_i|f(x_i), x_i) \backsim N(y_i|h(x_i)^T\beta + f(x_i), \sigma^2). \tag{5.4}$$

This is GPR model. Now we can say that this GPR model is non-parametric because for each x_i, a corresponding latent variable $f(x_i)$ introduced. When simplifying instance (5.4) to vector form, it is

$$P(Y|f, X) \backsim N(Y|H\beta + f, \sigma^2 I) \tag{5.5}$$

where

$$X = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \quad H = \begin{bmatrix} h(x_1^T) \\ h(x_2^T) \\ \vdots \\ h(x_N^T) \end{bmatrix}, \quad f = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_N) \end{bmatrix}.$$

For the latent variables $f(x_i)$ the joint distribution is

$$P(f|X) \backsim N(f|0, K(X, X)) \tag{5.6}$$

where $K(X, X)$ is a covariance function with form

$$K(X, X) = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_N) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N, x_1) & k(x_N, x_2) & \dots & k(x_N, x_N) \end{bmatrix}.$$

The covariance function can be kernel function or parametrized by hyper-parameters θ. If it is hypeparametrized, then $k(x, x')$ must be written as $k(x, x'|\theta)$. This means that kernel function is conditioned dependency on θ.

5.3 Kernel function

The purpose of kernel methods solutions is to build a module that performs the mapping into embedding of feature space and in machine learning a learning algorithm designed to discover linear patterns in that space. Kernel function is a mapping component which depends on data type. The algorithm is designed so that the real points is not needed, but only their pairwise inner

products. This is beneficial because these products can be computed directly from data items when using a kernel function (Definition 5.2). This kind of computing is very efficient. [10, 11]

Definition 5.2. A kernel function k satisfies for all $x, x' \in X$ so that

$$k(x, x') = \langle \phi(x), \phi(x') \rangle$$

where ϕ is a mapping function from X to feature space F

$$\phi : x \longrightarrow \phi(x) \in F.$$

In regression model, the kernel function can be used as covariance matrix. Kernel function can be expressed as $k(x, x'|\theta)$, where θ is vector containing the kernel parameters and it's terms can be parametrized. [10, 11]

When building a kernel function, a standard way is to use a standard deviation σ_f and a length scale σ_l to be the kernel parameters. The length scale defines maximum distance between inputs so that the response values become uncorrelated. Other parameter to building kernel function can be for example α which is a scale-mixture parameter. When using those two standard parameters as kernel parameters then we can define θ as

$$\theta = [\theta_1, \theta_2] = [\log(\sigma_l), \log(\sigma_f)], \quad \sigma_l, \sigma_f > 0. \tag{5.7}$$

The standard method is to use same length scale for all predictors, but it is not necessary. Some of standard methods use Euclidean distance (Definition 5.3) between two data points for defining kernel function.

Definition 5.3. Euclidean distance r for all $x, x' \in X$

$$r = \sqrt{(x - x')^T (x - x')}.$$

There is couple of the most commonly kernel functions and they are using the same length scale.

Squared Exponential Kernel

$$k(x, x'|\theta) = \sigma_f^2 \exp\left[-\frac{1}{2}\frac{(x-x')^T(x-x')}{\sigma_l^2}\right] \tag{5.8}$$

Exponential Kernel

$$k(x, x'|\theta) = \sigma_f^2 \exp\left[-\frac{-r}{\sigma_l}\right] \tag{5.9}$$

Matern 3/2

$$k(x, x'|\theta) = \sigma_f^2 \left(1 + \frac{\sqrt{3}r}{\sigma_l}\right) \exp\left(-\frac{\sqrt{3}r}{\sigma_l}\right) \tag{5.10}$$

Matern 5/2

$$k(x, x'|\theta) = \sigma_f^2 \left(1 + \frac{\sqrt{5}r}{\sigma_l} + \frac{5r^2}{3\sigma_l^2}\right) \exp\left(-\frac{\sqrt{5}r}{\sigma_l}\right) \tag{5.11}$$

Rational Quadratic Kernel

$$k(x, x'|\theta) = \sigma_f^2 \left(1 + \frac{r^2}{2\alpha\sigma_l^2}\right)^{-\alpha}, \quad \alpha > 0 \tag{5.12}$$

5.4 The computation of GPR method

Let (X, Y) be data where X is vector of input variables and Y is vector of response values. To predict these response values, we need to estimate

parameters in model (5.5). At first, we need to estimate parameters β, θ and σ^2.

For estimating parameters of model, we need to maximize the likelihood $P(Y|X)$ as function of parameters. The maximization problem can be written as

$$\hat{\beta}, \hat{\theta}, \hat{\sigma}^2 = \arg\max_{\beta,\theta,\sigma^2} \log\left[P(Y|X, \beta, \theta, \sigma^2)\right] \tag{5.13}$$

where $P(Y|X) = P(Y|X, \beta, \theta, \sigma^2) = N(y_i|H\beta, K(X, X'|\theta) + \sigma^2 I_N)$.

The marginal logarithm of likelihood function is

$$\begin{aligned}\log\left[P(Y|X, \beta, \theta, \sigma^2)\right] = &-\frac{1}{2}(y - H\beta)^T[K(X, X'|\theta) + \sigma^2 I_N]^{-1}(y - H\beta) \\ &- \frac{N}{2}log2\pi - \frac{1}{2}[K(X, X'|\theta) + \sigma^2 I_N].\end{aligned} \tag{5.14}$$

At first, we must compute estimate for function $\hat{\beta}(\theta, \sigma^2)$. For given θ and σ^2, the estimate of β maximizes log likelihood function (5.14)

$$\hat{\beta}(\theta, \sigma^2) = \left[H^T[K(X, X'|\theta + \sigma^2 I_N]^{-1}H\right]^{-1} H^T[K(X, X'|\theta + \sigma^2 I_N]y. \tag{5.15}$$

Now we can define the β-profiled log likelihood as

$$\begin{aligned}\log\left[P(Y|X, \hat{\beta}(\theta, \sigma^2), \theta, \sigma^2)\right] = &-\frac{1}{2}(y - H\hat{\beta}(\theta, \sigma^2))^T[K(X, X'|\theta) + \sigma^2 I_N]^{-1} \\ &(y - H\hat{\beta}(\theta, \sigma^2)) - \frac{N}{2}log2\pi - \frac{1}{2}[K(X, X'|\theta) \\ &+ \sigma^2 I_N].\end{aligned} \tag{5.16}$$

When estimate $\hat{\beta}$ is computed, then estimate of θ can be computed by maximizing β-profiled log likelihood over θ itself. Estimate of σ^2 can be computed similarly.

5.5 Prediction

Let predicted value of each variable be marked as $(\cdot)_p$. Definition of conditional probabilities is

$$P(f_p, f|X, x_p) = P(f_p|f, X, x_p) * P(f|X,x_p). \tag{5.17}$$

Probabilistic predictions of Gaussian Regression model are created from density $P(y_p|y, X, x_p)$. By definition of conditional probabilities (5.17), we can write density as

$$P(y_p|y, X, x_p) = \frac{P(y_p, y|X, x_p)}{P(y|X, x_p)}. \tag{5.18}$$

When using the latent variables of f and f_p, which are corresponding to y and y_p, we can compute density $P(y_p, y|X, x_p)$ as

$$\begin{aligned} P(y_p, y|X, x_p) &= \int\int P(y_p, y, f_p, f|X, x_p) df df_p \\ &= \int\int P(y_p, y|f_p, f, X, x_p) P(f_p, f|X, x_p) df df_p \end{aligned} \tag{5.19}$$

by using the joint distributions for same variables. We assume that each response values y_i depends on the latent variable f_i and the input x_i. Then $P(y_p, y|f_p, f, X, x_p)$ is a product of conditional densities and can be written as

$$P(y_p, y|f_p, f, X, x_p) = P(y_p|f_p, x_p) \prod_{i=1}^{N} P(y_i|f(x_i), x_i). \tag{5.20}$$

We know that result depend only on X and f and the formula of conditional probability is then

$$P(y|X, f) = \prod_{i=1}^{N} P(y_i|x_i, f_i) = P(y_p|x_p, f_p) \prod_{i=1}^{N} P(y_i|x_i, f(x_i). \tag{5.21}$$

Then we can rewrite the conditional density (5.19) as

$$P(y_p, y|X, x_p) = \int\int P(y_p|f_p, x_p) P(y|f, X) P(f_p|f, X, x_p) P(f|X, x_p) df df_p. \tag{5.22}$$

Also the density (5.20) can be re-expressed as

$$P(y_p, y|f_p, f, X, x_p) = P(y_p|x_p, f_p) P(y|X, f). \tag{5.23}$$

We know that

$$\begin{aligned} &P(f|X, x_p) = P(f|X) \\ &P(y|X, x_p) = P(y|X) \\ &P(y|X, f) P(f|X) = P(y, f|X) = P(f|y, X) P(y|X), \end{aligned} \tag{5.24}$$

then $P(y_p, y|X, x_p)$ (5.22) can be re-written as

$$P(y_p, y|X, x_p) = P(y|X) \int \int P(y_p|f_p, x_p) P(f|X, y) P(f_p|f, X, x_p) df df_p. \tag{5.25}$$

Hence, the density $P(y_p|y, X, x_p)$ (5.18) is

$$\begin{aligned} P(y_p|y, X, X_p) &= \frac{P(y_p, y|X, x_p)}{P(y|X, x_p)} = \frac{P(y_p, y|X, x_p)}{P(y|X)} \\ &= \int \int P(y_p|f_p, x_p) P(f|X, y) P(f_p|f, X, x_p) df df_p, \end{aligned} \tag{5.26}$$

where

$$\begin{aligned} &P(y_p|f_p, x_p) = N(y_p|h(x_p)^T\beta + f_p, \sigma_p^2) \\ &P(f|X, y) = N\left(f \middle| \frac{1}{\sigma^2}\left(\frac{I_N}{\sigma^2} + K(X, X')^{-1}\right)^{-1}(y - H\beta), \left(\frac{I_N}{\sigma^2} + K(X, X')^{-1}\right)^{-1}\right) \\ &P(f_p|f, X, x_p) = N\left(y_p|K(x_p^T, X)K(X, X')^{-1}f, k(x_p, x_p) - K(x_p^T, X)K(X, X')^{-1}K(X, x_p^T)\right). \end{aligned} \tag{5.27}$$

For a given new input x_p and the dataset (X, y), the density of the new response value y_p is

$$P(y_p|y, X, x_p) = N\left(y_p|h(x_p)^T\beta + \mu, \sigma_p^2 + \Sigma\right), \tag{5.28}$$

where $\mu = K(x_p^T, X)\left(K(X, X) + \sigma^2 I_N\right)^{-1}(y - H\beta)$ and $\Sigma = k(x_p, x_p) - K(x_p^T, X)K(X, X')^{-1}K(X, x_p^T)$.

The expected value of y_p can be calculate at new input x_p for the given dataset (X, y) and parameters β, θ and σ^2 with following formula

$$\begin{aligned} E[y_p|y, X, x_p, \beta, \theta, \sigma^2] &= h(x_p)^T\beta + K(x_p^T, X|\theta)\alpha \\ &= h(x_p)^T\beta + \sum_{i=1}^{N} \alpha_i k(x_p, x_i|\theta), \end{aligned} \tag{5.29}$$

where $\alpha = \left(K(X, X|\theta) + \sigma^2 I_N\right)^{-1}(y - H\beta)$.

6 Validation

The Lasso and GPR methods are used to determine the model between the input and output data. For training, I have used 1237 samples and for testing 557 samples. The number of input variables in our data was 9 and there were 14 target values. Because the dimension of the input variables have not been too big for using all of them to build the model, there wasn't any need to use dimensional reduction as preprocessing.

From experience, it is known that *input* 8 may be the most interesting input value. In addition, it is also known how it affects to the output variables. So, the sampling is done based on this knowledge and the sample is done by a weighted of input 8. The sampling has been so that there were a bit more samples of small and big values of input 8 and not so many values from the middle of the value range.

Different models are programmed using Matlab and its functions. The functions *lasso* and $fitrgp$ are available in Matlab to call Lasso regression and GRP algorithms. There is some simple example Matlab code for *lasso* in Appendix A and for $fitrgp$ in Appendix B. For the $fitrgp$ function it is possible to use different kernels. To find out the best model I used five different kernels. The point is that each kernel based on a different distribution. The distribution of the output variables is also changing and then it is important to use the best kernel option for a specific output. The used kernels were *Matern* 3/2 (m32), *Matern* 5/2 (m52), *exponential* (exp), *squared exponential* (se) and *rational quadratic* (rq) kernels. The form of those kernel functions have been presented in section 5.3 Kernel Function.

The accuracy of the model is calculated by the root mean squared error (RMSE, Definition 6.1.). This error tells the standard deviation of prediction errors. The prediction error is the distance between the predicted point and regression line. RMSE is calculated for testing data and I have used that value to choose the model for the outputs.

Definition 6.1. Root mean squared error for $y \in Y_{j,TEST}$ and $y_{pred} \in Y_{j,PRED}$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - y_{i,pred})^2}$$

where $j = 1, \ldots, R$ and R is number of outputs.

Models have been trained using different data than the testing data used for the model. Both data have been collected similarly for simulation. RMSE have measured to that testing data. Both training and testing data is normalized for same scale. Normalization for both is done by mean and standard deviation of training data for each variable. Then both data have the same value range in normalized scale.

For delta outputs (described in subsection 1.4 Data), there is margin selected, which defines how big delta value can be. Those margins are also normalized to same value range as training and testing data by mean and standard deviation of output training data. Margins are approximated for each output.

For each output variable, I used training data to do regression models. For each output there are built six different regression model, Lasso and GRP with five different kernels. When six models have done, I used evaluate data to predict RMSE value for each model. RMSE is calculated between original values and predicted values.

6.1 Results

Table 1 shows the RMSE values for each output with different models. We can see that for the output 12 all RMSE values are 0.000 which means that the all model fits perfect to that output. We can see also that for many outputs, RMSE values are same or almost the same for many GPR model with different kernels. For some outputs differences are very small. In that

Table 1: RMSE value for each output with both models Lasso and GPR.

	RMSE					
Output	Lasso	GPR m52	GPR m32	GPR exp	GPR se	GPR rq
y_1	0.6499	0.6128	0.6122	0.6135	0.6500	0.6165
y_2	0.8573	0.7877	0.7794	0.7671	0.8595	0.7901
y_3	0.5720	0.5861	0.5861	0.5861	0.5861	0.5861
y_4	1.0141	1.0122	1.0122	1.0122	1.0122	1.0122
y_5	0.7565	0.7535	0.7535	0.7535	0.7569	0.7535
y_6	1.0201	1.0236	1.0236	1.0236	1.0236	1.0236
y_7	0.9406	0.9445	0.9445	0.9445	0.9445	0.9445
y_8	3.1122	3.1150	3.1150	3.1150	3.1150	3.1150
y_9	0.7593	0.7691	0.7691	0.7691	0.7691	0.7691
y_{10}	0.9833	0.9864	0.9864	0.9864	0.9864	0.9864
y_{11}	1.6973	1.6971	1.6966	1.6952	1.6982	1.6939
y_{12}	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
y_{13}	0.8666	0.7969	0.7810	0.8666	0.8652	0.7857
y_{14}	1.0557	1.0586	1.0586	1.0586	1.0586	1.0586

case the choice between models is not very important, all will be predicting almost same points. Bigger distinctions are between outputs. For some outputs it is easier to build regression model than others. Models fit okay for outputs 1, 2, 3, 5, 7, 9 and 13. The rest of outputs has very badly fitting models. The mostly fitting models is Lasso and other models, which have been chosen to use are GPR with kernels *exponential*, *squared exponential*, *rational quadratic* and *matern*32. Summary of the best RMSE values and picked models for each output is shown in Table 2.

Table 3 shows intercept value and coefficients of each picked output models. These models are picked with best RMSE values. These RMSE values have been almost same for some models. Especially for Lasso models we see

Table 2: RMSE value for each output with best models.

Output	Best model	RMSE
y_1	Matern32	0.6122
y_2	Exponential	0.7671
y_3	Lasso	0.5720
y_4	SquaredExp	1.0122
y_5	RationalQuad	0.7535
y_6	Lasso	1.0201
y_7	Lasso	0.9406
y_8	Lasso	3.1122
y_9	Lasso	0.7593
y_{10}	Lasso	0.9833
y_{11}	RationalQuad	1.6939
y_{12}	Lasso	0.0000
y_{13}	Exponential	0.7810
y_{14}	Lasso	1.0577

that which input variable are the most meaningful variables for outputs. The useless coefficients for input variables are set to zero in models. For other models it is also tenable that the most important coefficients are the biggest and the useless ones are the smallest coefficients. The most common important input variables seem to be input 8 and input 2 based on coefficients β_2 and β_8. Also, inputs 7 and 9 are more important input values than the rest of inputs. For output 1 results are a bit different against other outputs. For this output the most important inputs are 1 and 4.

Figures about how regression models fit are in Appendix C. For all output there are two figures. The first shows original and predicted data points against input 8. The second figure show predicted points against reference line. This reference line beyond on original outputs. The purpose is that all predicted points should be on that reference line or even as close as possible.

For output 12 we see that all coefficients and intercept are zero, which

Table 3: Coefficients of best model for each output.

y_i	β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9
y_1	0.01	-0.64	-0.05	-0.03	0.44	-0.01	-0.06	-0.02	-0.01	0.01
y_2	-0.26	-0.18	-0.05	0.05	0.07	0.01	0.08	-0.07	0.67	0.14
y_3	-1.6e-15	0	-0.08	0	0	0	0	0	-0.16	0
y_4	3.9e-04	-0.01	-0.09	0.04	0.04	0.01	0.01	0.05	-0.20	0.04
y_5	3.6e-04	0.02	-0.02	0.05	0.03	0.08	0.01	0.03	-0.12	0.04
y_6	-5.5e-16	0.07	-0.11	-0.01	0	0	0.01	0.05	-0.20	0.02
y_7	7.2e-16	0	-0.05	0	0	0.002	0	0	-0.22	0
y_8	1.7e-15	0	-0.08	0	0	-0.02	0	0	-0.16	0
y_9	1.4e-15	0	-0.08	0	0	0	0	0	-0.18	0
y_{10}	-3.2e-16	0	-0.05	0	0	0	0	0	-0.10	0
y_{11}	9.4e-04	0.002	-0.04	0.08	0.04	0.03	0.07	0.05	-0.08	0.08
y_{12}	0	0	0	0	0	0	0	0	0	0
y_{13}	0.06	-0.33	3.5e-04	-0.05	0.26	-0.04	-0.19	0.02	0.43	-0.05
y_{14}	2.9e-17	0	0	0	0	0	0	0	0	0

means that all values of outputs are zero in training data also. In this case it means that for output 12 delta between flp and fxp value is always zero and there is no need to concentrate to this output anymore. Also, for output 14 all coefficients are set to zero. Only intercept value is bigger than zero, but it is very small value.

For special inspection, output 2 is very important. It tells how many UEs are connected to broadband network. From coefficient tables we can see coefficients of output 2 model. The biggest coefficient value is for input 8 which explain most that is UE able to connected broadband network or not. The next biggest coefficients are β_1. We can see from Figure 5 that in very low input 8 value there is no any UE connect to network. The limit is around -1 in normalized scale. When predicted values is more than -0.5 in normalized scale then it is rounded to 0 in same scale and it means that UE

is connected to network.

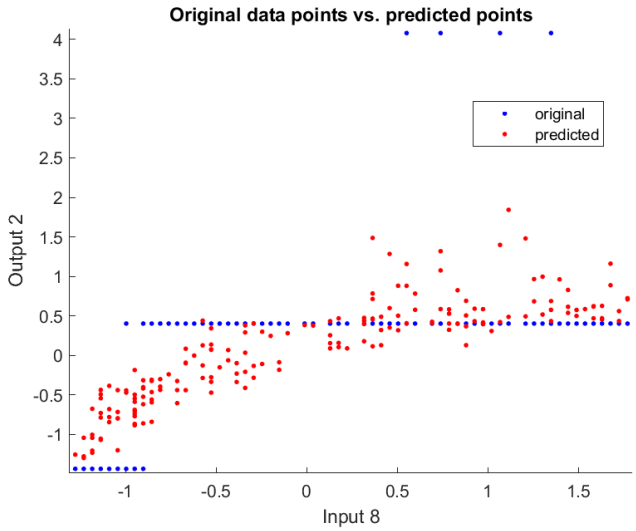


Figure 5: Original vs predicted points for regression model of output 2.

Outputs from 3 to 14 are delta value between flp and fxp. In ideal situation this should be zero like output 12 is. For optimizing simulation behaviour my purpose was to find output variables which have mostly large delta values. Means that they are not zero. We have to approximate margins for these deltas. We can see from Figure 6 that there are no values even near that margin for output 14. This means that we can also forget that output like 12.

For many delta outputs there are almost all data points very small like the purpose have defined. However, there are a few values which are very much higher than others. We know that those are not outliers and then cannot be removed from data. Those data points have very big effect for the regression model. Because of those data points RMSE value is very high for those outputs which have that kind of data. We can see from Figure 7 example of that kind of output. For these kinds of outputs, it is very difficult to build regression model or any kind of model. From figures in Appendix C we can see that input 8 explain quite much those big values. For small values

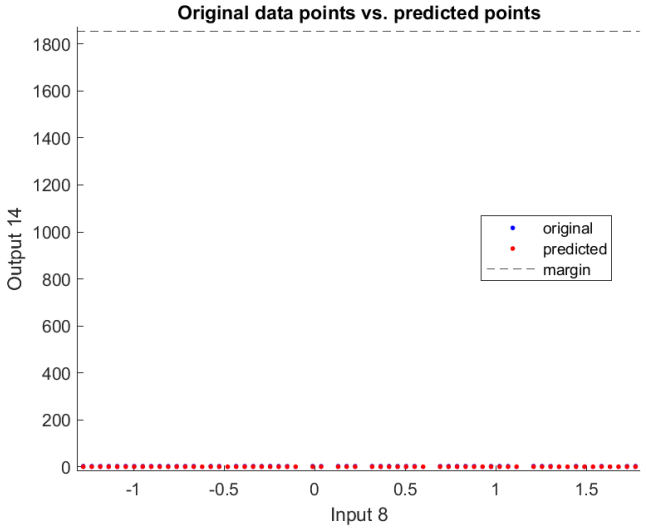


Figure 6: Original vs predicted points for regression model of output 14.

of input 8 there could be big values for some outputs. Those big values are very much higher than margin is. This means that these input combination cases should be in closer inspection in the future.

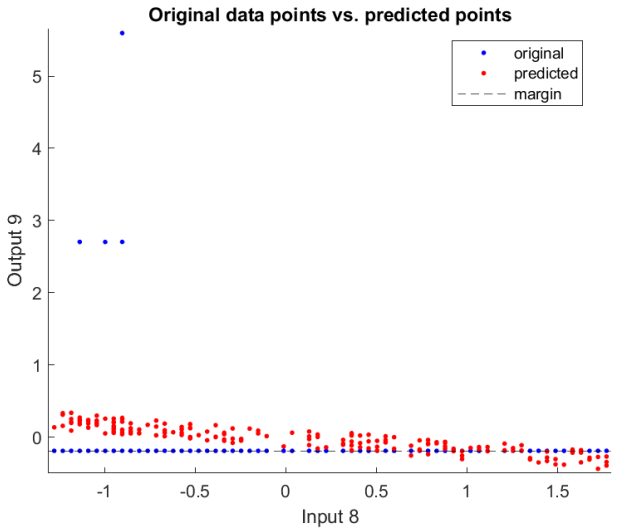


Figure 7: Original vs predicted points for regression model of output 9.

For output 10 there are also a lot of big values for high input 8 value. Figure 8 shows that those values are still under margin with some space.

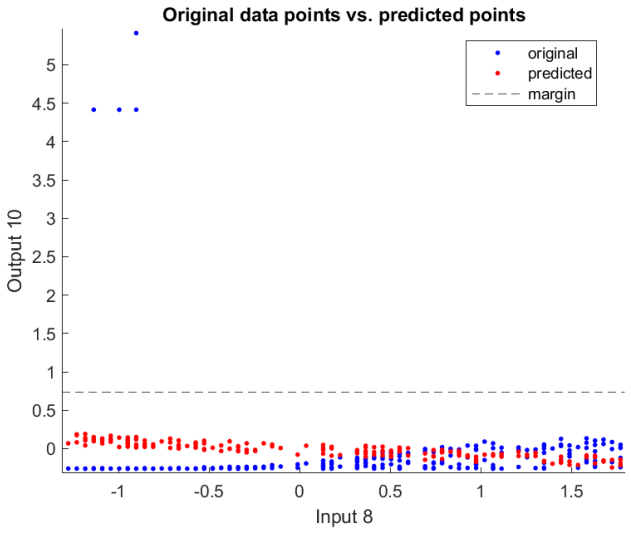


Figure 8: Original vs predicted points for regression model of output 10.

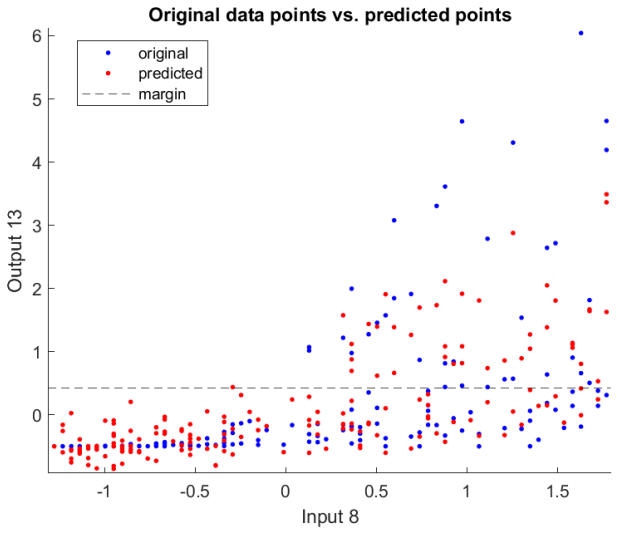


Figure 9: Original vs predicted points for regression model of output 13.

For output 13 there are a lot of data points which are higher than margin. Input 8 contributes much to these data points. When this input value has bigger than -0.5 value in normalized scale, there is chance to get too big output value. The bigger value of the input 8 cause usually bigger output value. From coefficient tables we can see that also inputs 1,4 and 6 explains those output values. Also input 4 have positive affect to increase value but 1 and 6 decrease that. Figure 9 shows original and predicted data point for output 13 against input 8.

6.2 Discussion

We can see from results that for most outputs it is very difficult to build good regression model. For some outputs there was only a few other than zero values that model can learn them easily. Those big values attract the reference line for themselves. We still know that those big values belong to data and cannot be removed out of it. The volume of data should have been bigger that the possibility to learn those big values would increase.

There was margin for each delta outputs. For some outputs, the predicted delta values get higher values than margin is. The reason is that those big values in training data have so big effect for learned regression model. In truth, most delta values are less than margin in training and testing data and that should be the case also in predicted data. This does not happen for most outputs because the model is not fitting for them. On the other way, predicted model does not catch up those big value when predicting data points. Ideal situation would be that the model finds out those worst cases from input combinations.

Both selected methods are based on a linear combination. Since the starting point of the methods is quite the same, the models do not differ much either. A small difference can be found for some output models. Because this point of view clearly does not work well, it is needed to use some non-linear regression models to get better results.

7 Summary

Demonstrating data-driven simulated PRACH data can be very challenging because of the output data. When the most interesting data is a delta value between flp and fxp values and the expected values are zero for all outputs, it is very difficult to build regression models for that kind of data. In this situation there is a lot of zeros and small values in the output data and a very low percent of output values are big values. When building a regression model for the data, the big values have a huge impact on the regression line, and they are increasing the root mean squared errors.

The machine learning algorithms Lasso and GPR are good methods for building regression models between input and output data if the connection is linear. There were not big differences between the models for the PRACH data. For both methods, most of models were not very suitable for predicting. Machine learning algorithms would need more training to build those models.

In the data, there were multivariate aspects. There were 9 inputs and 14 outputs. The number of input variables was so little that there was no need to use any variable selection method for reducing the dimension. Of course, the Lasso method is doing that naturally and that showed us the most important variables in the models. With that size of inputs, the machine learning algorithms were fast enough and not causing any troubles for the training data. We decided to build our own model for each output variable. All outputs were independent results from the PRACH simulation, so the best options to demonstrate the data were to construct some of my own multivariate regression models for each output. With that kind of situations handling, we were also able to compare more easily the output behaviours between each other.

The goal was to try to demonstrate regression models regarding the PRACH data and investigate the input and output variables. The first research question that needed to be investigated was how input variables influence outputs. Most of the models did not fit well and the affects were hard to find out. With those models it is not possible to predict new target

values reliably. Two of the three research questions concerned variables that is it possible to find out the most meaningful input variables and whether the output variables with the most high delta values from the data. When we compared the results to the questions, we found inputs which are the most affecting for some outputs, and we know a few outputs which have the biggest delta values.

8 Futureworks

The purpose of this thesis was to investigate PRACH data to optimize simulator behaviour. The results of predicted regression models using Machine Learning algorithms Lasso and GPR are not suitable for this goal. With those methods we can find some most meaningful input and output variables, but we cannot predict output variables reliably.

In that situation for future there are two different ways to continue this investigating. First, it is possible to find other methods to build those models. A non-linear regression model can be useful at least for some outputs. One possible non-linear method is logistic regression which uses a dichotomic response to build the model. For building models, there are other possibilities than machine learning to do that. For example, Neural Networks or decision trees can be useful. Anyway, it can be difficult to find other building methods which can handle the challenging situation of PRACH data and delta values.

The second method is to analyze PRACH data without building models between input and outputs variables. That kind of analyzing method could be for example PCA or anomaly detection. With PCA, it is possible to find variables which explain most of variances in data. By anomaly detection we can possibly find out the worst cases about the input combinations. The purpose would be to find some useful information from data that variables in simulator can be analyzed and optimized.

References

[1] Rajat Harlalka. Choosing the Right Machine Learning Algorithm. [Online; accessed 7.11.2019]. 2018. URL: https://hackernoon.com/choosing-the-right-machine-learning-algorithm-68126944ce1f

[2] Ian Goodfellow, Youshua Bengio and Aaron Courville. Deep Learning. @ 2016 An MIT Press book.

[3] O'REILLY. ElasticNet regression. [Online; accessed 28.2.2020] URL:https://www.oreilly.com/library/view/machine-learning-with/9781787121515/5c5ec380-d139-49a5-99b1-3ce32ae5bd6f.xhtml

[4] MathWorks. Exact GPR Method. [Online; accessed 24.2.2020]. URL: https://se.mathworks.com/help/stats/exact-gpr-method.html

[5] Jere Puolakanaho. Digital Twin using multivariate Prediction. Pro Gradu, University of Oulu, 2019.

[6] C. E. Rasmussen, K. I. Williams. Gaussian Processes for Machine Learning. Massachusetts Institute of Technology, ISBN 026218253X. @ 2006 MIT Press.

[7] MathWorks. Gaussian Process Regression Models. [Online; accessed 30.1.2020]. URL: https://se.mathworks.com/help/stats/gaussian-process-regression-models.html

[8] E. C. Alexopoulos. Introduction to Multivariate Regression Analysis. PMCID: PMC3049417. @ 2010 Hippokratia.

[9] Gordan Žitkovič. Introduction to Stochastic Processes. @ 2010 The University of Texas at Austin.

[10] MathWorks. Kernel (Covariance) Function Options. [Online; accessed 31.1.2020]. URL: https://se.mathworks.com/help/stats/kernel-covariance-function-options.html

[11] John Shawe-Taylor, Nello Cristianini. Kernel Methods for Pattern Analysis. @ 2004 Cambridge University Press.

[12] Statistics How To. Lasso Regression: Simple Definition. [Online; accessed 11.11.2019]. 2015. URL: https://www.statisticshowto.datasciencecentral.com/lasso-regression/

[13] Fraser J. Forbes and Ilyasse Aksikas. Lecture Notes on Engineering Optimization. [Online; accessed 1.11.2019]. 2002-2005. URL: https://sites.ualberta.ca/ aksikas/nlpm.pdf

[14] Donglin Wang, Raja R.Sattiraju, Anjie Qiu, Sanket Partani, Hans D. Methodologies of Link-Level Simulator and System-Level Simulator for C-V2X Communication. Schotten. @ 2019 VTC Fall workshop'19.

[15] A.C. Rencher. Methods of Multivariate Analysis. @ 2002 Wiley & Sons.

[16] J.McNames. Multivariate Optimization. [Online; accessed 4.11.2019]. URL: http://web.cecs.pdx.edu/ edam/Slides/MultivariateOptimizationx4.pdf

[17] Statistics How To. Nonlinear Regression: Simple Definition & Examples. [Online; accessed 30.1.2020]. URL: https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/multivariate-analysis/

[18] Seelan Sundaralingam. PRACH (gNB-ABIL) L1 feature specification. @ 2019 NOKIA. (not available)

[19] William Heidkamp. Predicting the concentration of residual methanol in industrial formalin using machine learning. Bachelor Degree Project, Karlstads Universitet, 2016.

[20] MegaM@art 2. Project Overview. [Online; accessed 1.11.2019]. 2019. URL: https://megamart2-ecsel.eu/overview/

[21] Frank E. Harrell, Jr. Regression Modeling Strategies. @ 2015 Springer.

[22] Trevor Hastie, Robert Tibshirani, Martin Wainwright. Statistical Learning with Sparsity; The Lasso and Generalizations. @ 2015 CRC Press.

[23] Trevor Hastie, Robert Tibshirani and Jerome Friedman. The Elements of Statistical Learning; Data Mining, Inference, and Prediction. @ 2017 Springer.

[24] Saleh Aldran, Ahmad Alshammari, Mohammad Matin. Uplink channel estimation error for large scale MIMO system. Conference Paper, Conference: SPIE Optical Engineering + Applications, DOI: 10.1117/12.2238004. 2016.

[25] Stefan Pratschner, Bashar Tahir, Ljiljana Marijanovic, Mariam Mussbah, Kiril Kirev, Ronald Nissel, Stefan Schwarz, Markus Rupp. Versatile mobile communications simulation: the Vienna 5G Link Level Simulator. EURASIP Journal on Wireless Communications and Networking (2018) 2018:226.

[26] 3GPP Organizational Partners. 3GPP TS 38.213 Chapter 8. @ 2019 NOKIA. (not available)

Appendix A

In Matlab there is 'Statistics and Machine Learning Toolbox' and inside that there is function names as 'lasso'. Here is example code how to use it in Matlab.

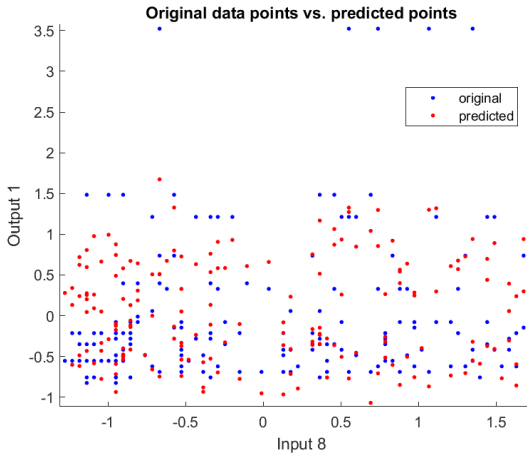
```
% Normalize data
X_train_stand = normalize(X_train);
X_test_stand = normalize(X_test);
Y_train_stand = normalize(Y_train);
Y_test_stand = normalize(Y_test);

% Fitting Lasso
[B,FitInfo] = lasso(X_stand,Y_stand,'CV',10,...
  'PredictorNames','x1','x2','x3','x4','x5', 'x6','x7','x8','x9',...
  'MCReps', 5);
idxLambdaMinMSE = FitInfo.IndexMinMSE;

% plot Cross-Validaation plot
lassoPlot(B,FitInfo,'PlotType','CV');
legend('show')

% predict Yhat using model
coef = B(:,idxLambdaMinMSE);
coef0 = FitInfo.Intercept(idxLambdaMinMSE);
Yhat = X_test * coef + coef0;

% compute RMSE and plot results
RMSE = sqrt(mean((Y_test - Yhat).^2));
figure();
hold on;
title('Original data points vs.  predicted points');
scatter(Y_test);
plot(Yhat, 'ro');
legend('original','predicted', 'Location', 'best');
hold off;

figure();
hold on;
title('Predicted values vs.  actual exam grades against a reference line');
scatter(Y_test,Yhat);
plot(Y_test,Y_test);
xlabel('Actual Exam Grades');
ylabel('Predicted Exam Grades');
legend('feature point','reference line', 'Location', 'northwest');
hold off;
```

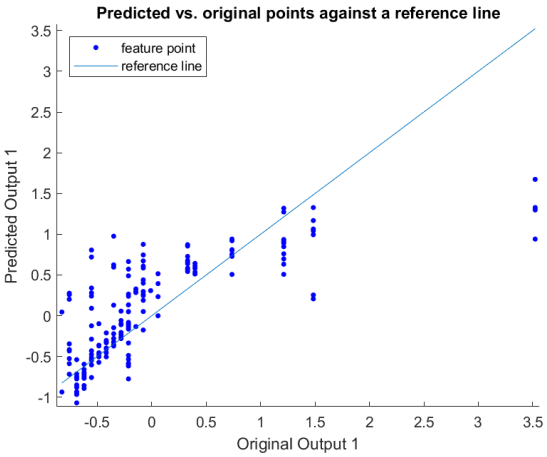
Appendix B

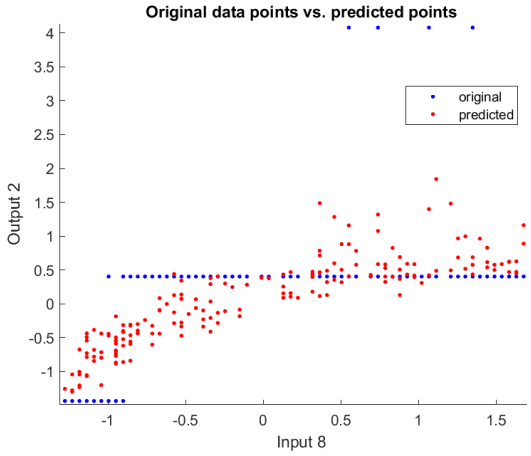
To use GPR in Matlab, you need same toolbox and function named as 'fitrgp'. Here is a example code for it.

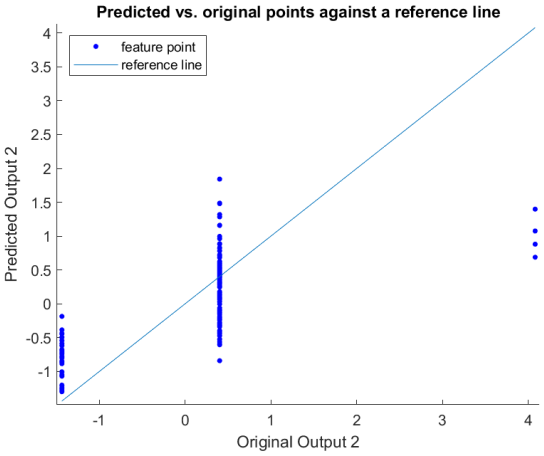
```
% Normalize data
X_train_stand = normalize(X_train);
X_test_stand = normalize(X_test);
Y_train_stand = normalize(Y_train);
Y_test_stand = normalize(Y_test);

% Fitting GPR
model_rgp = fitrgp(X,Y, 'Kernelfunction','Matern52',...
  'FitMethod','exact','PredictMethod','exact',...
  'Basis','linear');

% predict ypred using model
ypred = predict(model_rgp, X_test);

% compute RMSE and plot results
RMSE = sqrt(mean((Y_test - ypred).^2));
figure();
hold on;
title('Original data points vs.  predicted points');
scatter(Y_test);
plot(ypred, 'ro');
legend('original','predicted', 'Location', 'best');
hold off;

figure();
hold on;
title('Predicted values vs.  actual exam grades against a reference line');
scatter(Y_test,ypred);
plot(Y_test,Y_test);
xlabel('Actual Exam Grades');
ylabel('Predicted Exam Grades');
legend('feature point','reference line', 'Location', 'northwest');
hold off;
```

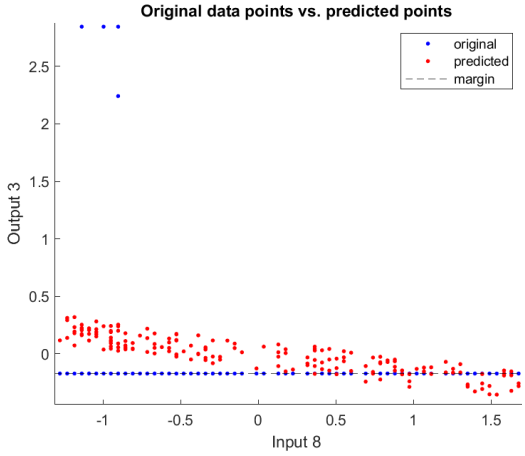
Appendix C

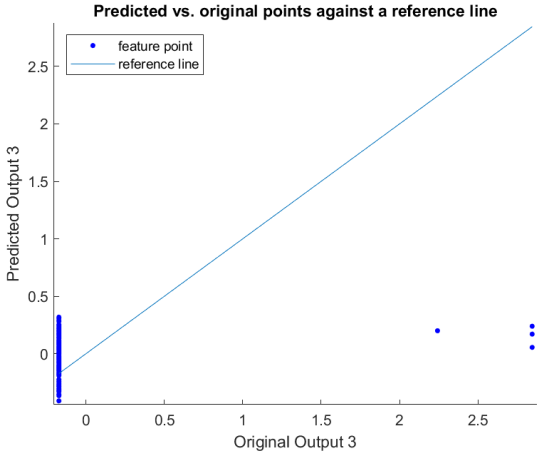

Original data points vs. predicted points
original
predicted
Output 1
Input 8

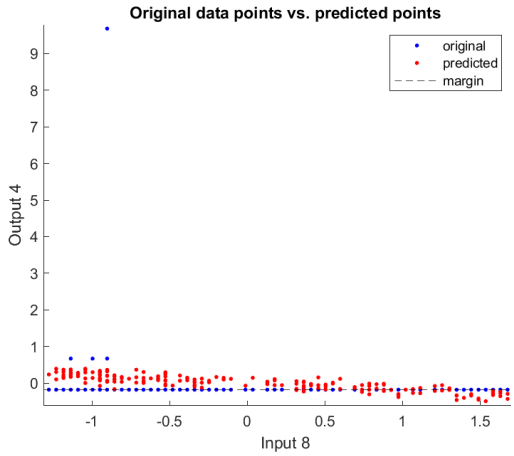

Predicted vs. original points against a reference line
feature point
reference line
Predicted Output 1
Original Output 1

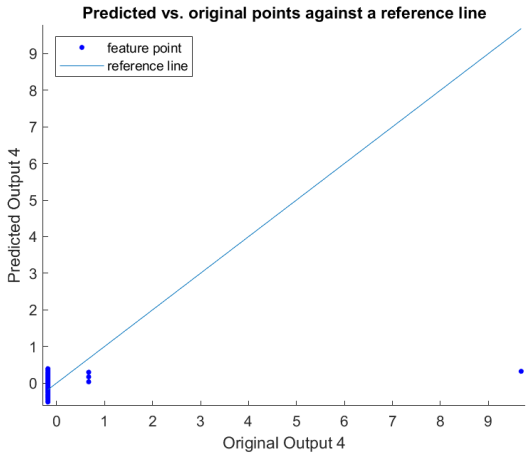

Original data points vs. predicted points
original
predicted
Output 2
Input 8

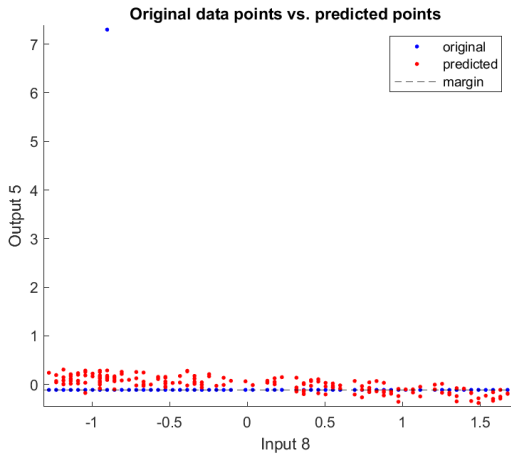

Predicted vs. original points against a reference line
feature point
reference line
Predicted Output 2
Original Output 2

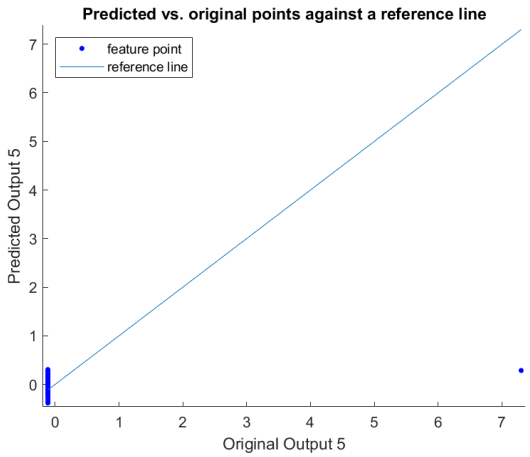

Original data points vs. predicted points
original
predicted
margin
Output 3
Input 8

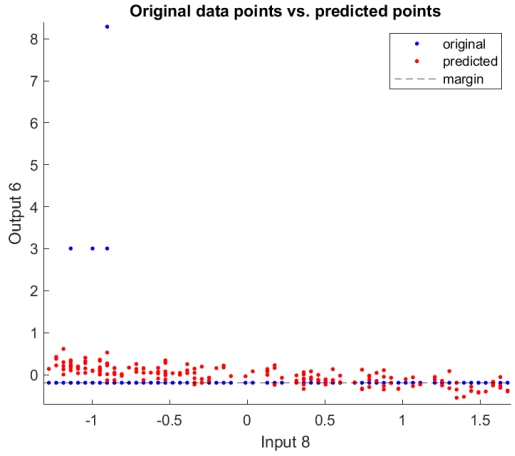

Predicted vs. original points against a reference line
feature point
reference line
Predicted Output 3
Original Output 3

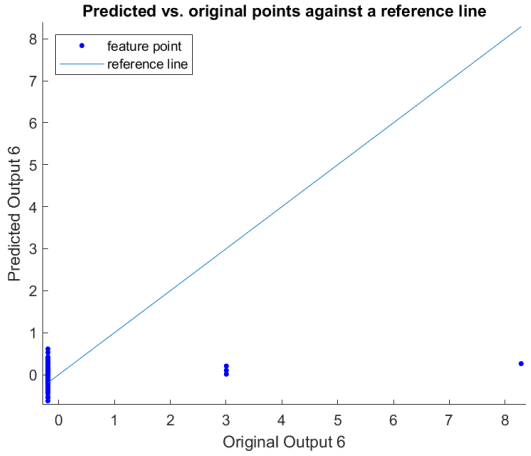

Original data points vs. predicted points
original
predicted
margin
Output 4
Input 8

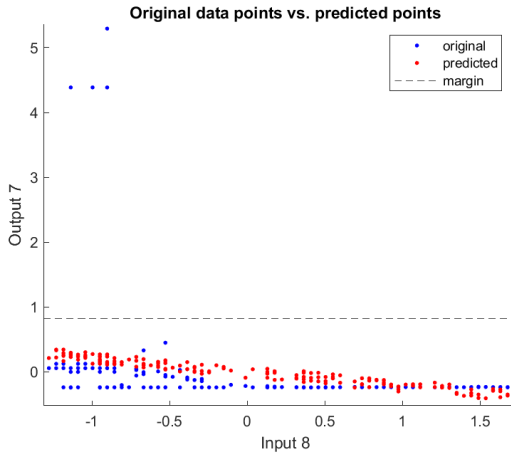

Predicted vs. original points against a reference line
feature point
reference line
Predicted Output 4
Original Output 4

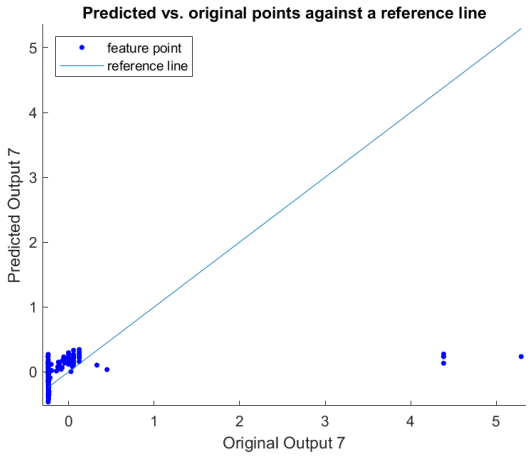

Original data points vs. predicted points
original
predicted
margin
Output 5
Input 8

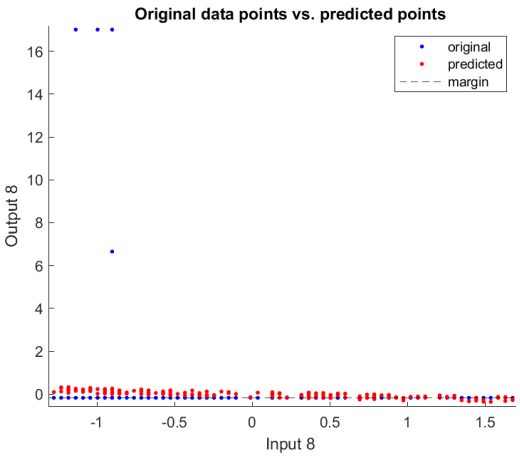

Predicted vs. original points against a reference line
feature point
reference line
Predicted Output 5
Original Output 5

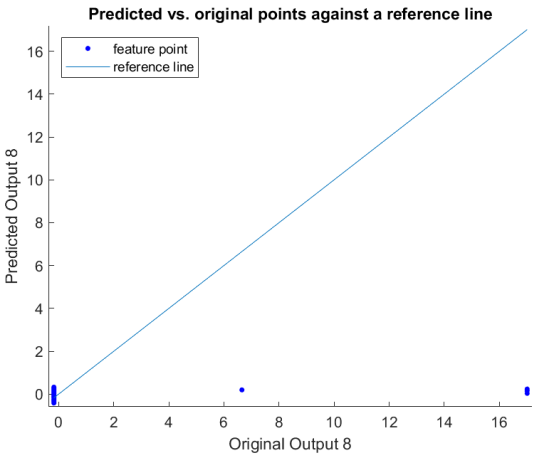

Original data points vs. predicted points
original
predicted
margin
Output 6
Input 8

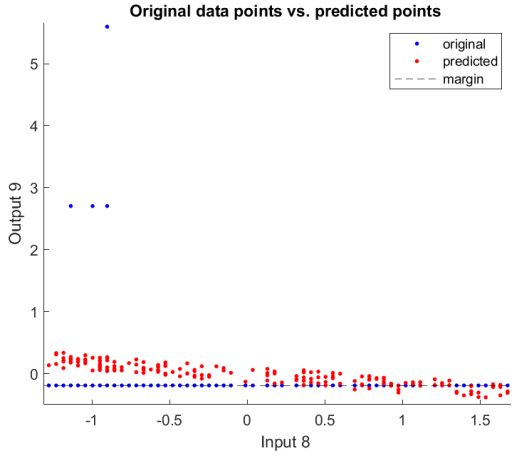

Predicted vs. original points against a reference line
feature point
reference line
Predicted Output 6
Original Output 6

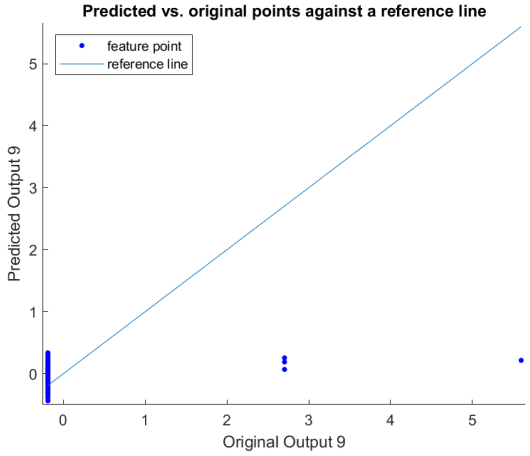

Original data points vs. predicted points
original
predicted
margin
Output 7
Input 8

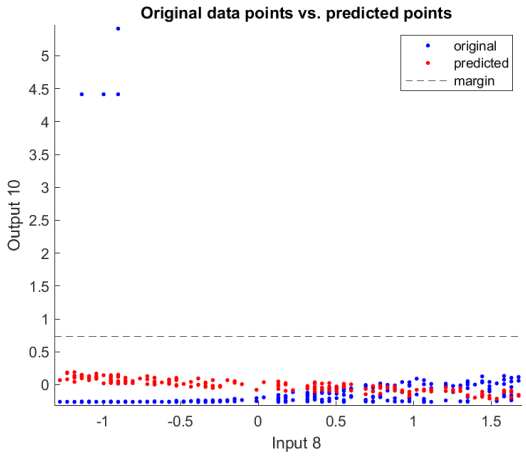

Predicted vs. original points against a reference line
feature point
reference line
Predicted Output 7
Original Output 7

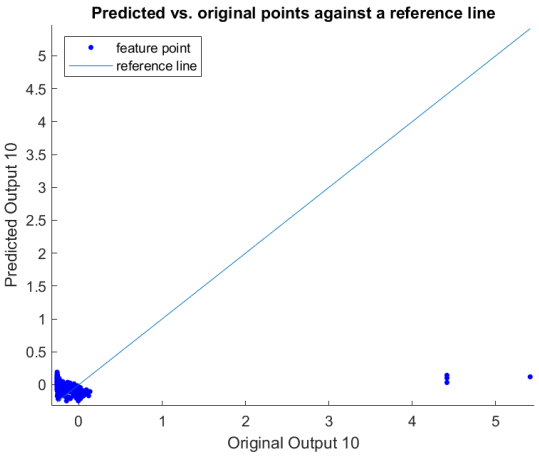

Original data points vs. predicted points
original
predicted
margin
Output 8
Input 8

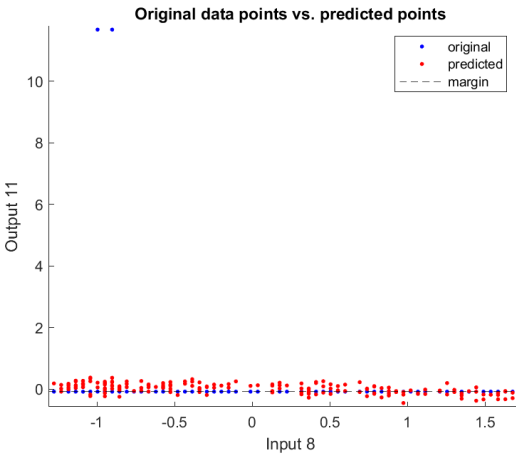

Predicted vs. original points against a reference line
feature point
reference line
Predicted Output 8
Original Output 8

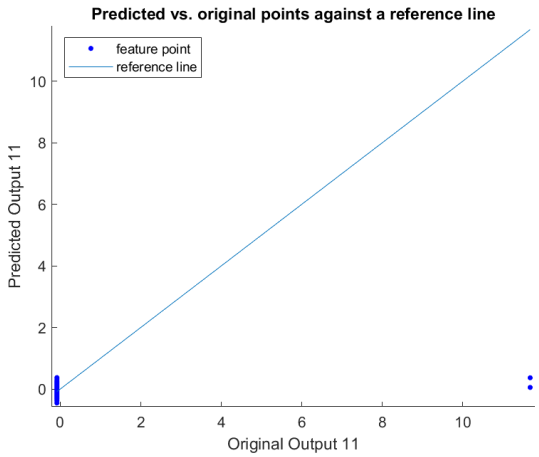

Original data points vs. predicted points
original
predicted
margin
Output 9
Input 8

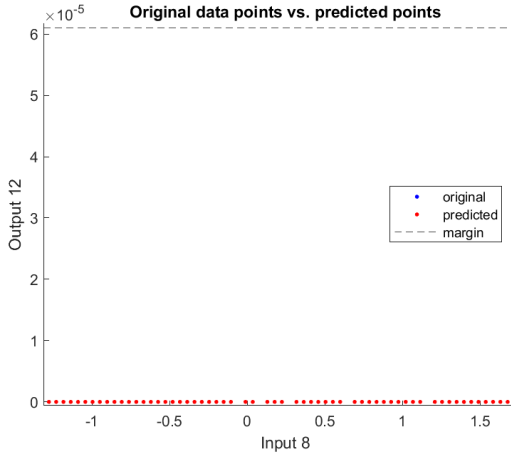

Predicted vs. original points against a reference line
feature point
reference line
Predicted Output 9
Original Output 9

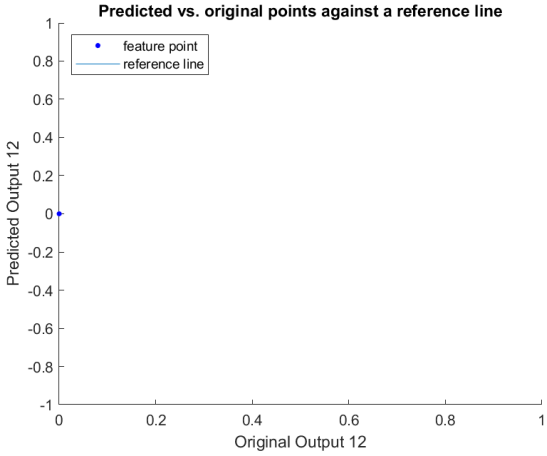

Original data points vs. predicted points
original
predicted
margin
Output 10
Input 8

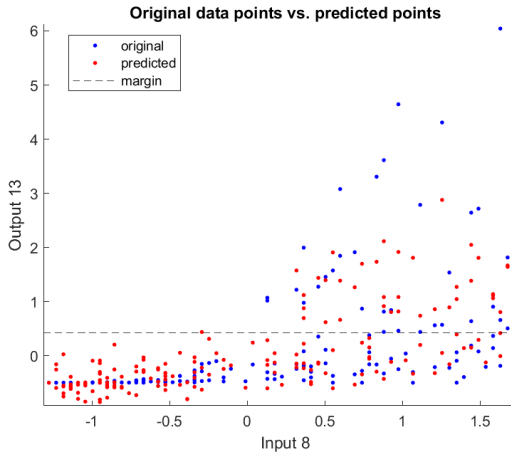

Predicted vs. original points against a reference line
feature point
reference line
Predicted Output 10
Original Output 10

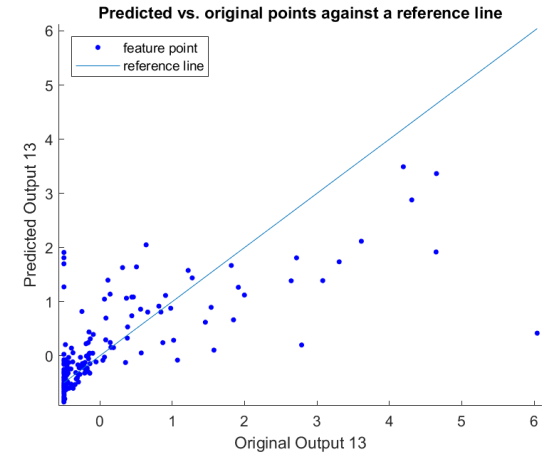

Original data points vs. predicted points
original
predicted
margin
Output 11
Input 8

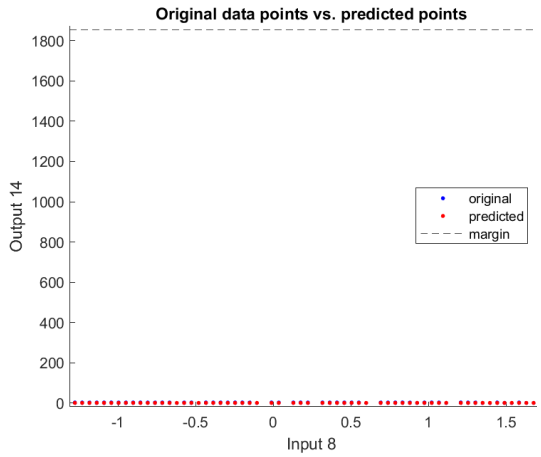

Predicted vs. original points against a reference line
feature point
reference line
Predicted Output 11
Original Output 11

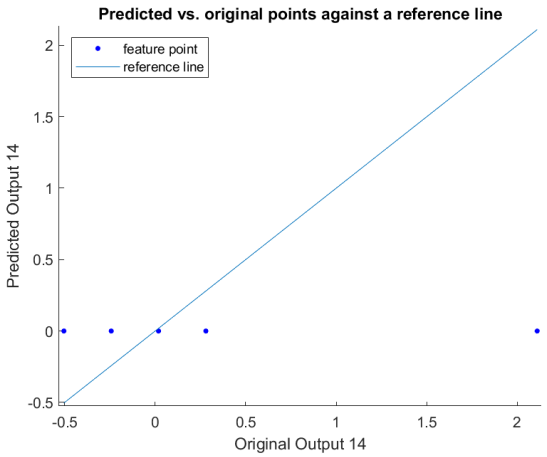

Original data points vs. predicted points
×10^-5
Output 12
Input 8
original
predicted
margin


Predicted vs. original points against a reference line
feature point
reference line
Predicted Output 12
Original Output 12


Original data points vs. predicted points
Output 13
Input 8
original
predicted
margin


Predicted vs. original points against a reference line
feature point
reference line
Predicted Output 13
Original Output 13


Original data points vs. predicted points
Output 14
Input 8
original
predicted
margin


Predicted vs. original points against a reference line
feature point
reference line
Predicted Output 14
Original Output 14